**Q1-------------------------**

**Single inheritance**

Write a Java program to implement Single Inheritance.  
  
**[Note: Strictly adhere to the object oriented specifications given as a part of the problem statement. Use the same class names and member variable names.  
Follow the naming conventions mentioned for getters/setters]**   
  
Consider a class named **Person**with the following private data members.

|  |  |
| --- | --- |
| **Data Type** | **Data Member** |
| String | name |
| String | dateOfBirth |
| String | gender |
| String | mobileNumber |
| String | bloodGroup |

Include appropriate getters and setters.  
  
Consider a class named **Donor**which extends **Person**class with the following private data members.

|  |  |
| --- | --- |
| **Data Type** | **Data Member** |
| String | bloodBankName |
| String | donorType |
| String | donationDate |

Include appropriate getters and setters.  
  
The class **Donor**should have the following method

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void displayDonationDetails( ) | This method displays the donation details. Display the statement ‘Donation Details :’ inside this method |

Consider another class **Main**and write the main method to test the above class.  
  
In the main( ) method, read the person and donor details from the user and call the displayDonationDetails( ) method.  
  
**[Note:** The date format should be **“dd-MM-yyyy”]**  
  
  
  
**Input and Output Format:**  
  
Refer sample input and output for formatting specifications.  
**All text in bold corresponds to input and the rest corresponds to output.**  
  
**Sample Input and Output 1:**

Enter the name :  
**Justin**  
Enter Date of Birth :  
**11-01-1995**  
Enter Gender :  
**Male**  
Enter Mobile Number :  
**9994910354**  
Enter Blood Group :  
**B+ve**  
Enter Blood Bank Name :  
**Blood Assurance**  
Enter Donor Type :  
**Whole Blood**  
Enter Donation Date :  
**09-07-2017**  
Donation Details :  
Name : Justin  
Date Of Birth : 11-01-1995  
Gender : Male  
Mobile Number : 9994910354  
Blood Group : B+ve  
Blood Bank Name : Blood Assurance  
Donor Type : Whole Blood  
Donation Date : 09-07-2017  
  
  
**Sample Input and Output 2:**

Enter the name :

**Steffie**

Enter Date of Birth :

**12-01-1996**

Enter Gender :

**Female**

Enter Mobile Number :

**8868875432**

Enter Blood Group :

**O+ve**

Enter Blood Bank Name :  
**Edward Blood Bank**

Enter Donor Type :

**Whole Blood**

Enter Donation Date :

**21-12-2016**  
Donation Details :

Name : Steffie

Date Of Birth : 12-01-1996

Gender : Female

Mobile Number : 8868875432

Blood Group : O+ve

Blood Bank Name : Edward Blood Bank

Donor Type : Whole Blood

Donation Date : 21-12-2016

Q2------------------------------------------

**Calculate Reward Points**

ABC Bank announced a new scheme of reward points for a transaction using an ATM card. Each transaction using the normal card will be provided 1% of the transaction amount as a reward point. If a transaction is made using a premium card and it is for fuel expenses, additional 10 points will be rewarded. Write a java program to calculate the total reward points.  
  
**[Note:  Strictly adhere to the object-oriented specifications given as a part of the problem statement.  
Follow the naming conventions as mentioned]**  
  
Consider a class **VISACard**with the following method.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public Double computeRewardPoints(String type, Double amount) | This method returns the 1% of the transaction amount as reward points |

Consider a class **HPVISACard**which extends **VISACard** class and overrides the following method.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public Double computeRewardPoints(String type, Double amount) | In this method, calculate the reward points from the base class and add 10 points if it is for fuel expense |

**Hint:**  
  
Use super keyword to calculate reward points from base class.  
  
Consider **the Main**class with the**main** method and read all the transaction details in the main method.  
Card type will be either **‘VISA card’ or ‘HPVISA card’**. Otherwise, display **‘Invalid data’**  
  
Calculate the reward points corresponding to the card type and transaction type and print the reward points(upto two decimal places).  
  
  
**Input and Output Format:**  
  
Refer sample input and output for formatting specifications.  
Enter the transaction details in CSV format ( Transaction type, amount, card type)  
  
**All text in bold corresponds to the input and the rest corresponds to output.**  
  
**Sample Input and Output 1:**  
  
Enter the transaction detail  
**Shopping,5000,VISA card**  
Total reward points earned in this transaction is 50.00  
Do you want to continue?(Yes/No)  
**Yes**  
Enter the transaction detail  
**Fuel,5000,HIVISA card**  
Invalid data  
Do you want to continue?(Yes/No)  
**Yes**  
Enter the transaction detail  
**Fuel,5000,HPVISA card**  
Total reward points earned in this transaction is 60.00  
Do you want to continue?(Yes/No)  
**No**  
  
**Sample Input and Output 2:**  
  
Enter the transaction detail  
**Fuel,1000,HPVISA card**  
Total reward points earned in this transaction is 20.00  
Do you want to continue?(Yes/No)  
**No**

**GST Calculation**

Write a program to calculate the total amount with GST for the events. The Events are Stage show and Exhibition. For the Stage show, GST will be 15% and for exhibition, GST will be 5%.  
  
**[Note: Strictly adhere to the object oriented specifications given as a part of the problem statement.  
Follow the naming conventions as mentioned]**  
  
Consider class **Event** with the following protected attributes/variables.

|  |  |
| --- | --- |
| **Data Type** | **Variable** |
| String | name |
| String | type |
| Double | costPerDay |
| Integer | noOfDays |

Prototype for the parametrized constructor,  
**public Event(String name, String type, Double costPerDay, Integer noOfDays)**  
  
Consider class **Exhibition** which extends the **Event** class with the following private attributes/variables.

|  |  |
| --- | --- |
| **Data Type** | **Variable** |
| static Integer | gst = 5 |
| Integer | noOfStalls |

Prototype for the parametrized constructor,  
**public Exhibition(String name, String type, Double costPerDay, Integer noOfDays, Integer noOfStalls)**  
  
Include appropriate getters and setters.  
  
Include the following method.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public Double totalCost() | This method is to calculate the total amount with 5% GST |

Consider class **StageEvent** which extends the **Event** class with the following private attributes/variables.

|  |  |
| --- | --- |
| **Data Type** | **Variable** |
| static Integer | gst = 15 |
| Integer | noOfSeats |

Prototype for the parametrized constructor,  
**public StageEvent(String name, String type, Double costPerDay, Integer noOfDays, Integer noOfSeats)**  
  
Include appropriate getters and setters.  
  
Include the following method.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public Double totalCost() | This method is to calculate the total amount with 15% GST |

**Override toString()**method in all classes to display the event details in the format specified in sample input and output.  
  
Consider **Main** class with **main** method.  
In the main() method, read the event details from the user and then create the object of the event according to the event type.  
The total amount will be calculated according to the GST of the corresponding event.  
  
  
**Input and Output Format:**  
  
Refer sample input and output for formatting specifications.  
All the double values should formatted to two decimal places.  
  
**All text in bold corresponds to the input and the rest corresponds to output.**  
  
**Sample Input and Output 1:**  
  
Enter event name  
**Sky Lantern Festival**  
Enter the cost per day  
**1500**  
Enter the number of days  
**3**  
Enter the type of event  
1.Exhibition  
2.Stage Event  
**2**  
Enter the number of seats  
**100**  
Event Details  
Name:Sky Lantern Festival  
Type:Stage Event  
Number of seats:100  
Total amount: 5175.00

**Sample Input and Output 2:**  
  
Enter event name  
**Glastonbury**  
Enter the cost per day  
**5000**  
Enter the number of days  
**2**  
Enter the type of event  
1.Exhibition  
2.Stage Event  
**1**  
Enter the number of stalls  
**10**  
Event Details  
Name:Glastonbury  
Type:Exhibition  
Number of stalls:10  
Total amount: 10500.00  
  
**Sample Input and Output 3:**  
  
Enter event name  
**Glastonbury**  
Enter the cost per day  
**5000**  
Enter the number of days  
**2**  
Enter the type of event  
1.Exhibition  
2.Stage Event  
**3**  
Invalid input

Top of Form

Bottom of Form

Q4-----------------------------------------

**Abstract Class**

Write a program to calculate total cost of the event based on the type of event and display details using Abstract class and method.  
  
**Strictly adhere to the Object-Oriented specifications given in the problem statement. All class names, attribute names and method names should be the same as specified in the problem statement.**  
  
Consider an **abstract** class called **Event** with following protected attributes.

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| name | String |
| detail | String |
| type | String |
| organiser | String |

Prototype for the parametrized constructor, **Event(String name, String detail, String type, String organiser)**  
Include appropriate getters and setters

Include the following abstract method in the class Event.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| abstract Double calculateAmount() | an abstract method |

Consider a class named **Exhibition**which extends **Event** class with the following private attributes

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| noOfStalls | Integer |
| rentPerStall | Double |

Prototype for the parametrized constructor,  
**public Exhibition(String name, String detail, String type, String organiser, Integer noOfStalls,Double rentPerStall)**  
Include appropriate getters and setters

Use super( ) to call and assign values in base class constructor.

Include the following abstract method in the class Exhibition.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| Double calculateAmount () | This method returns the product of noOfStalls and rentPerStall |

Consider a class named **StageEvent** which extends **Event**class with the following private attributes.

|  |  |
| --- | --- |
| **Attribute** | **Datatype** |
| noOfShows | Integer |
| costPerShow | Double |

Prototype for the parametrized constructor,  
**public StageEvent(String name, String detail, String type, String organiser, Integer noOfShows,Double costPerShow)**  
Include appropriate getters and setters

Use super( ) to call and assign values in base class constructor.

Include the following abstract method in the class StageEvent.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| Double calculateAmount() | This method returns the product of noOfShows and costPerShow |

Consider a driver class called **Main**. In the main method, obtain input from the user and create objects accordingly.  
  
  
**Input format:**  
Input format for Exhibition is in the CSV format (**name,detail,type,organiser,noOfStalls,rentPerStall)**  
Input format for StageEvent is in the CSV format (**name,detail,type,organiser,noOfShows,costPerShow)**      
  
**Output format:**  
Print "**Invalid choice**" if the input is invalid to our application and terminate.  
Display one digit after the decimal point for Double datatype.  
Refer to sample Input and Output for formatting specifications.  
  
**[All text in bold corresponds to the input and rest corresponds to output]**  
  
**Sample Input and output 1:**

Enter your choice

1.Exhibition

2.StageEvent

**1**

Enter the details in CSV format

**Book expo,Special sale,Academics,Martin,100,1000**

Exhibition Details

Event Name:Book expo

Detail:Special sale

Type:Academics

Organiser Name:Martin

Total Cost:100000.0

**Sample Input and Output 2:**

Enter your choice

1.Exhibition

2.StageEvent

**2**

Enter the details in CSV format

**JJ magic show,Comedy magic,Entertainment,Steffania,5,1000**

Stage Event Details

Event Name:JJ magic show

Detail:Comedy magic

Type:Entertainment

Organiser Name:Steffania

Total Cost:5000.0

**Sample Input and Output 3:**

Enter your choice

1.Exhibition

2.StageEvent

**3**

Invalid choice

 Overriding is a runtime polymorphism. The inherited class has the overridden method which has the same name as the method in the parent class. The argument number, types, or return types should not differ in any case. The method is invoked with the object of the specific class ( but with the reference of the parent class).  
  
Write a program to calculate projected revenue for exhibition and stage event using inheritance and method overriding  
  
**[Note :  
Strictly adhere to the object-oriented specifications given as a part of the problem statement.  
Use the same class names and member variable names. ]**  
Consider a parent class **Event** and define the following protected attributes,

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| name | String |
| detail | String |
| ownerName | String |

Include appropriate **getters and setters**.  
Prototype for the parameterized constructors to the Event class in the following order **Event(String name, String detail, String ownerName)**

Declare the abstract method**public abstract Double projectedRevenue()** in the Event class  
  
Consider a child class **Exhibition** that extends **Event**  that defines with the following attribute,

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| noOfStalls | Integer |

Include appropriate **getters and setters**.  
Prototype for the parameterized constructors to the Exhibition class in the following order **Exhibition(String name, String detail, String ownerName, Integer noOfStalls).**Use super( ) to call and assign values in the base class constructor.

Implement the abstract method **projectedRevenue()** in Exhibition class

|  |  |
| --- | --- |
| **MethodName** | **Description** |
| public Double projectedRevenue() | Calculate revenue and return the double value. Each stall will produce Rs.10000 as revenue |

Consider another child class **StageEvent** that extends Event that defines with the following attribute

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| noOfShows | Integer |
| noOfSeatsPerShow | Integer |

Include appropriate **getters and setters**.  
Prototype for the parameterized constructors to the **StageEvent** class in the following order **StageEvent(String name, String detail, String ownerName, Integer noOfShows, Integer noOfSeatsPerShow).**Use super( ) to call and assign values in the base class constructor.

Implement the abstract method **projectedRevenue()** in StageEvent class

|  |  |
| --- | --- |
| **MethodName** | **Description** |
| public Double projectedRevenue() | Calculate revenue and return the double value. Each seat produces Rs.50 revenue. |

Consider the class Main. It includes the method main. In the main( ) method  the event details are read from the user and the methods of the above classes are called  
  
     
**Input and Output Format:**  
  
Refer to sample input/output for other further details and format of the output.  
The double values should be formatted to 1 decimal place.  
  
**[All Texts in bold corresponds to the input and rest are output]**  
  
**Sample Input/Output 1:**  
  
Enter the name of the event:  
**Science Fair**  
Enter the detail of the event:  
**Explore Technology**  
Enter the owner name of the event:  
**ABCD**  
Enter the type of the event:  
1.Exhibition  
2.StageEvent  
**1**  
Enter the number of stalls:  
**65**  
The projected revenue of the event is 650000.0  
  
**Sample Input/Output 2:**  
  
Enter the name of the event:  
**Magic Show**  
Enter the detail of the event:  
**See Magic without Logic**  
Enter the owner name of the event:  
**SDFG**  
Enter the type of the event:  
1.Exhibition  
2.StageEvent  
**2**  
Enter the number of shows:  
**10**  
Enter the number of seats per show:  
**100**  
The projected revenue of the event is 50000.0